

## Petits projets en python

Pour aborder cette partie, il convient d'avoir toujours à portée de main l'aide mémoire sur les commandes usuelles de python

### Exercice 1

Ecrivez un programme qui fait saisir un prix hors taxes à l'utilisateur, puis qui affiche le prix TTC correspondant en se basant sur un taux de TVA à 19,6%.

→ Utilisez des variables aux noms explicites.

→ la demande d'une information saisie au clavier par un utilisateur est codée par input("la question posée")

Modifiez ensuite votre programme pour permettre à l'utilisateur de choisir le taux de TVA.

### Exercice 2

On donne un extrait des logins d'accès au réseau du lycée :

Identifiant	Mot de passe
s.levy	mon_password
j.etienne	0123456
p.rey	joli_caillou
j.bernon	essap_ed_toM

1. Créer le dictionnaire "acces" ayant pour clés les différents identifiants et les mots de passe correspondant en valeur

2. Créer une fonction "identification()" qui demande à l'utilisateur de saisir son identifiant et son mot de passe (grâce à input()) et retourne "True" si l'identification est bonne ou "False" si elle est incorrecte

Aide : "nom\_de\_cle" in dictionnaire.keys() permet de savoir si "nom\_de\_cle" est présente dans le dictionnaire.

### Exercice 3

Ecrire un programme qui, étant donné une équation du second degré  $a \cdot x^2 + b \cdot x + c = 0$ , détermine le nombre de ses solutions réelles et leurs valeurs éventuelles.

Rappel : le discriminant du trinôme  $a \cdot x^2 + b \cdot x + c$  est  $\Delta = b^2 - 4 \cdot a \cdot c$

Si  $\Delta < 0$  alors l'équation n'admet aucune solution réelle.

Si  $\Delta = 0$  alors l'équation admet une solution double :  $x = \frac{-b}{2 \cdot a}$

Si  $\Delta > 0$  alors l'équation admet 2 solutions réelles distinctes :

$$x_1 = \frac{-b - \sqrt{\Delta}}{2 \cdot a} \quad \text{et} \quad x_2 = \frac{-b + \sqrt{\Delta}}{2 \cdot a}$$

### Exercice 4 : D'une chaîne de caractères à une image

Il est traité dans une certaine mesure une chaîne de caractères comme une liste : si on définit **chaîne="voiture"** et que l'on demande **chaîne[2]** cela retournera 'i'. De même **len(chaîne)** retournera 7. on ne peut cependant pas modifier une lettre de la même façon qu'on modifie un item d'une liste.

La chaîne suivante

**chaîne2="001111000100001010100101100000110100101100110010100001000111100"**

encode une image qu'il va vous falloir décoder : le 0 correspond à du blanc codé par un carré blanc char(9608) et 1 a un carré noir codé char(9617). L'image fait 8 pixel de largeur.

Niveau 1 : transformer cette chaîne2 en liste où chaque item est un caractère

Niveau 2 : transformer cette chaîne2 en une liste de listes où chaque sous-liste correspond à une ligne de l'image

Niveau 3 : Dessiner l'image à l'aide de la fonction **print()**

Niveau 4 : Ecrire une fonction **dessiner(chaîne, largeur)** qui dessine l'image à l'aide de la fonction print() en fonction de la largeur annoncée de l'image

Niveau 12 : Modifier la fonction **dessiner(chaîne, largeur)** pour dessiner directement l'image sans passer par la création d'une liste